IES de Teis
Avda. de Galicia, 101
36216 – Vigo

Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

Unidad didáctica 1. Hadoop.

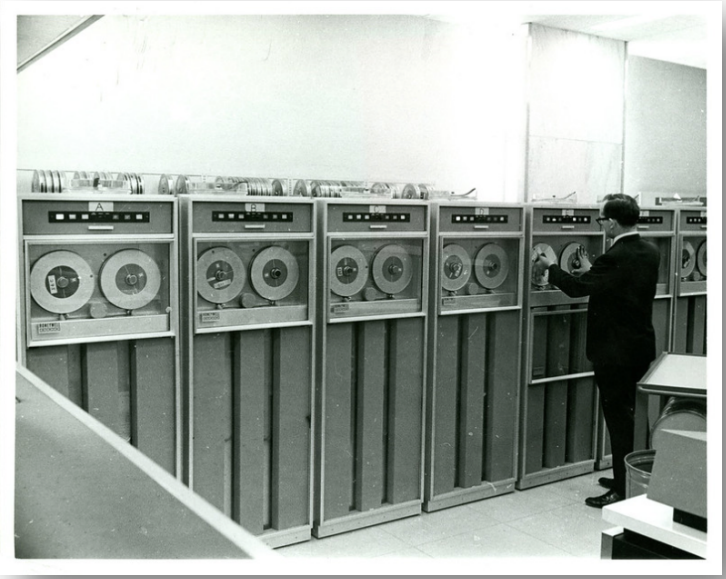
IES de Teis
Avda. de Galicia, 101
36216 – Vigo

Tfno: 886 12 04 64
e-mail: ies.teis@edu.xunta.es
http://www.iesteis.es

Tabla de contenido

1.- Origen

En la década de 1970 los estados, bancos y grandes compañías manejaban una cantidad importante de datos que guardaban de forma casi secuencial en equipos especializados.



La llegada de los semiconductores permitió que el hardware diera un salto de gigante propiciando mejoras en almacenamiento, proceso y comunicaciones. A nivel de software Edgar Codd estableció los fundamentos de las bases de datos relacionales permitiendo que fuera sencillo unificar, relacionar y consultar todos los datos. Esta posibilidad de almacenar mucha información supuso el inicio de los ERPs o software de planificación de recursos empresariales como por ejemplo SAP.

En la década de 1990, Tim Berners-Lee crea la Word Wide Web y con ella llega el crecimiento exponencial de generación de datos. Los datos ya no solo se producen y consultan en las grandes empresas, cualquiera puede hacerlo desde su casa.

Internet supuso un nuevo impulso para la generación de datos. Algunas empresas como Yahoo en 1996 vieron potencial en crear un directorio de páginas web y otros servicios. Más tarde, en 1998, nace Google con una función similar pero una filosofía distinta, el motor de búsqueda.

A principios del siglo XXI llega la web 2.0 donde es usuario ya no es solo un consumidos de información y ahora puede interactuar y generar contenido. Entre los años 2005 y 2006 nacen wikis, foros, blogs, Youtube y redes sociales como Facebook que permitieron la generación de contenido como nunca se había visto.

Algunas empresas se dieron cuenta que los datos eran valiosos ya que podían generar patrones, tendencias y que incluso se podían usar para influir en las decisiones de personas. Google, Microsoft y Amazon dedicaron muchos recursos al almacenamiento y gestión de los datos.

La llegada del Internet de las Cosas y la cantidad de datos que genera hizo que el problema de la gestión y almacenamiento de los datos tuviera que ser afrontado desde otro punto de vista. Google dedica muchos recursos a estudiar este problema y llega a las siguientes conclusiones:

- El hardware falla.
- Los archivos son de tamaño muy grande.
- La mayoría de los archivos crece añadiendo secuencialmente nuevo contenido.

Con estas premisas y basándose en el paradigma de computación distribuida crea y publica en 2003 la solución que sentará las bases de Big Data. Por un lado, crea GFS (Google File System) un sistema de ficheros distribuido escalable y confiable. También define el modelo MapReduce en el que los nodos, además de almacenar información pueden procesarla. Por último, también crean y definen BigTable como un sistema de almacenamiento y consulta para datos estructurados.

Entre 2006 y 2008, Doug Cutting y Mike Cafarella , usan las publicaciones de Google para desarrollar un proyecto para Yahoo relacionado con aumentar la velocidad de las búsquedas de resultados. Este proyecto dio origen a Hadoop.

Desde el 2008 Hadoop es un proyecto de código abierto, pero es en 2012 cuando la Apache Software Foundation se hace cargo de su gestión, mantenimiento y puesta a disposición pública.

Nacen compañías como Cloudera o Hortonworks que se encargan de proporcionar Hadoop a las empresas y cobrar por su soporte. En muchas ocasiones estas compañías crean software que se usa ampliamente y más allá de sus propias distribuciones.

2.- ¿Qué es Apache Hadoop?

Una buena definición de Hadoop, según el creador del mismo Doug Cutting, es que Hadoop es el kernel para Big Data.

Hablar de Hadoop es hablar de una suite de programas o un entorno de trabajo con el que programar y ejecutar aplicaciones distribuidas. De la misma manera que una suite ofimática como Microsoft Office no es un solo programa, Hadoop tampoco lo es.

La filosofía de Hadoop es divide y vencerás. Los trabajos de almacenamiento, así como los de procesamiento se dividen en porciones más pequeñas para poder tratarlas de manera distribuida y simultánea haciendo que todas las operaciones sean órdenes de magnitud más rápidas.

Hay problemas que bien por su tamaño o bien por su manera de resolver no son adecuados para un entorno distribuido. En una pequeña suma de varios números de pocas cifras la distribución del cálculo tiene mucho más sobrecoste en todos los sentidos que dedicar un único recurso.

Usar Apache Hadoop proporciona un sistema de almacenamiento en nodos distribuidos llamado **HDFS** que veremos en profundidad en el tema 2. Por ahora será suficiente con decir que este sistema de ficheros se expande horizontalmente y proporciona alto rendimiento y fiabilidad. En principio nada nuevo, los sistemas de ficheros distribuidos ya existían desde hace tiempo.

La novedad de Hadoop es que permite el uso de esos nodos no solo para almacenar sino también para procesar. La filosofía anterior consistía en llevar los datos desde el almacenamiento al equipo procesador en cambio con Hadoop el código se ejecuta en el mismo lugar que se guardan los datos. Para ello es necesario establecer un modelo llamado **MapReduce** que se encargará de procesar los datos de un nodo y de fusionarlos con los datos de los restantes nodo.

Almacenar y procesar datos en el mismo nodo exige algún planificador de recursos, el de Hadoop se llama **YARN**. En temas siguiente veremos con detalle su funcionamiento.

Por último, Hadoop incorpora **bibliotecas comunes** que sirven de soporte para otras aplicaciones del ecosistema Hadoop.

3.- Ecosistema Hadoop y distribuciones

El ecosistema Hadoop está formado por todas las aplicaciones que hacen uso de almacenamiento y procesamiento distribuido.

Hadoop trabaja con miles de nodos susceptibles de estropearse y por tanto parece natural que aparezca alguna herramienta de monitorización como Ambari.

La distribución de datos y capacidad de proceso hace de este entorno algo interesante para bases de datos que aprovecharían el mayor espacio y la mayor velocidad en consultas, Cassandra o Hbase son dos ejemplos.

Importar datos de fuentes diversas a Hadoop también es una utilidad contemplada dentro del ecosistema con la aplicación Sqoop.

Dedicaremos un tema en exclusiva para probar y clasificar algunas de las herramientas de Hadoop.

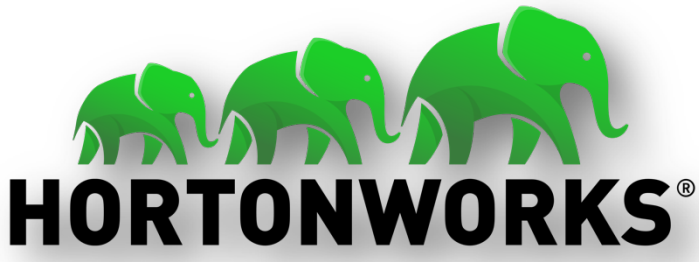
En general las herramientas del ecosistema Hadoop comparten el mismo problema, están gestionadas como proyectos independientes de la Apache Software Foundation. Esto hace que evolucionen por caminos diversos haciendo que cumplir las dependencias de todas las herramientas a la vez sea muy complicado o imposible con algunas combinaciones.

Las distribuciones basadas en Hadoop seleccionan las aplicaciones compatibles, dan soporte y en muchas ocasiones acompañan con herramientas propias que proporcionan un valor añadido para las empresas que quieran implantar una solución Hadoop.

Las distribuciones Hadoop más conocidas han sido:

- **Cloudera**. Fue la primera distribución basada en Hadoop liderada por el propio Dough Cutting. Esta distribución se caracteriza por su innovación constante. Algunas de sus aplicaciones como por ejemplo Impala se consideran plenamente del entorno Hadoop.

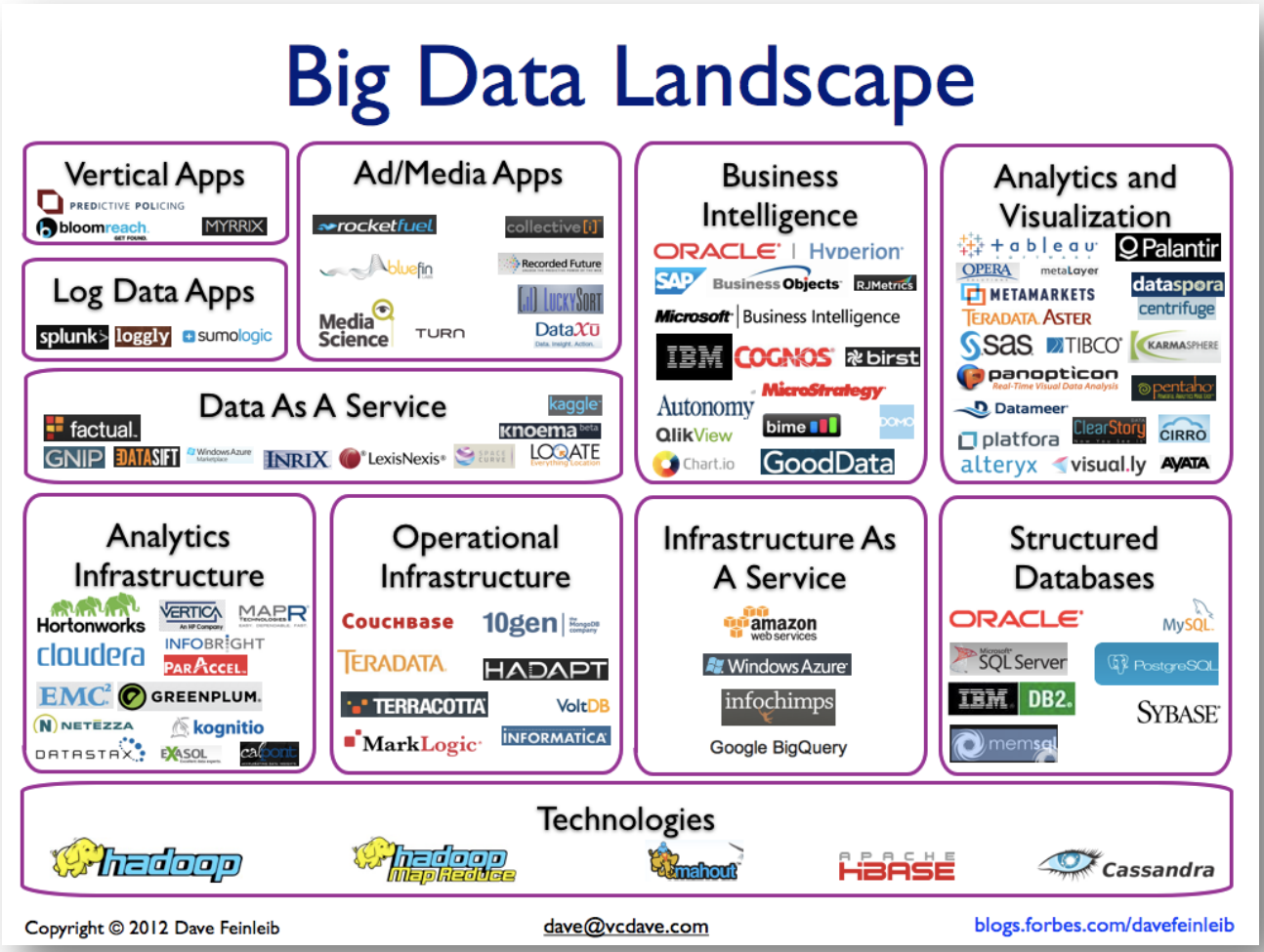
- **Hortonworks.** Esta distribución surge como respuesta a Cloudera por parte de Yahoo. Se caracteriza por ser 100% Apache Open Source. En esta suite también encontramos herramientas propias pero que se consideran plenamente del entorno Hadoop como por ejemplo Ambari.



- **MapR**. La característica más representativa de esta distribución es que se centra en la parte de almacenamiento HDFS hasta el punto de crear un sistema de ficheros propio donde prima la alta disponibilidad.

En la actualidad el panorama ha cambiado debido a que Cloudera ha absorbido a Hortonworks. Fruto de esta fusión ha quedado Cloudera como una distribución única con las herramientas de una y otra. Además, desde el 2021, Cloudera requiere de una suscripción para poder acceder a sus productos.

Por otro lado, el ecosistema Hadoop crece fuera de las distribuciones con herramientas y aplicaciones de uso específico. En el siguiente gráfico se muestra el estado del arte en el año 2012. Aquí se pueden ver herramientas de Hadoop pero también otras como herramientas de inteligencia de negocio, visualización o bases de datos. Buscando en Google por “big data landscape” encontrarás actualizaciones más recientes donde se incluyen muchas más nuevas herramientas surgidas durante estos años.

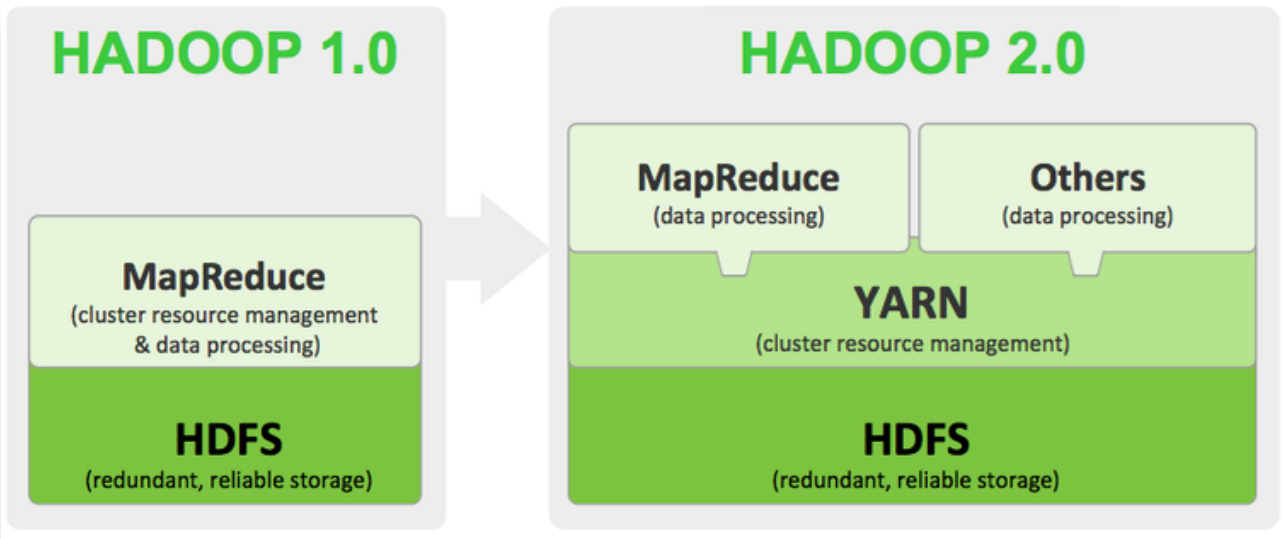


4.- Arquitectura

En el año 2012 se publicaba la versión 1 de Hadoop contando únicamente con HDFS y MapReduce. Un año más tarde se publica la versión 2 que incluye como novedad el gestor de recursos YARN.

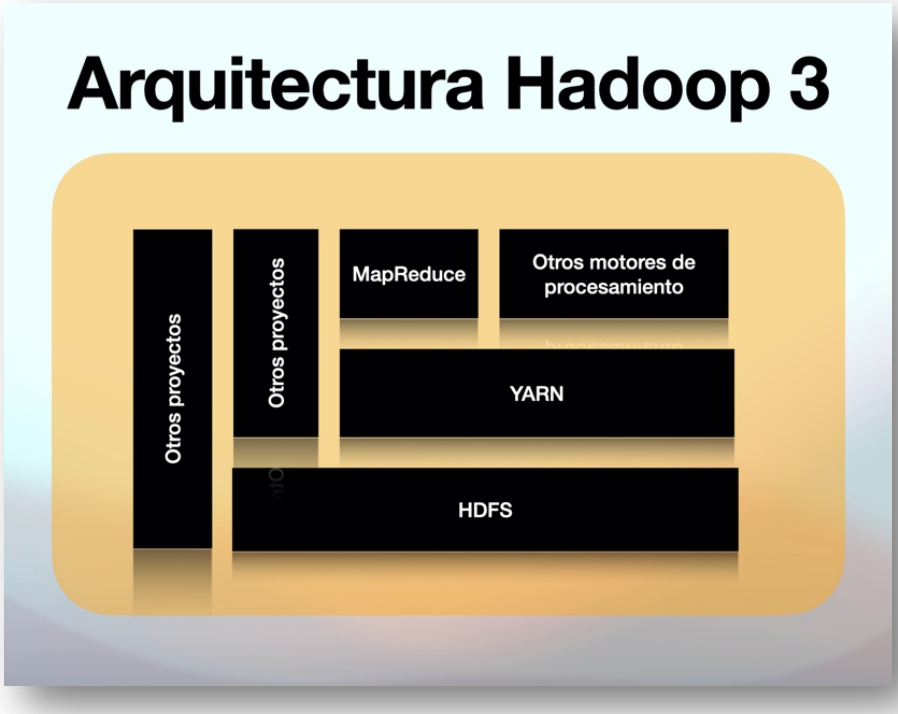
La diferencia fundamental es que en la primera versión de Hadoop existían solo dos capas. En la capa de almacenamiento el sistema de ficheros HDFS permitía la distribución de ficheros entre nodos. Estos nodos también permitían procesar datos que estuvieran en ese nodo. Para lo relacionado con el proceso únicamente teníamos MapReduce.

Rápidamente se detectaron problemas cuando la carga de trabajo era grande. Los recursos estaban ocupados y las nuevas tareas que llegaban no podían ejecutarse. Como respuesta a este problema, la versión 2 de Hadoop incluye



una capa intermedia llamada YARN. Esta capa es realmente un framework que soporta varios motores de ejecución además de MapReduce. YARN también es el encargado de planificar y monitorizar los trabajos que se encuentran en ejecución y los recursos necesarios para ello.

En el 2017 se publica Hadoop versión 3 que incluye mejoras como un servidor incorporado de DNS, soporte para Java 11, soporte de contenedores y mejoras en la API referente a HDFS.



La base de Hadoop es la capa HDFS, encargada fundamentalmente del almacenamiento y el proceso distribuido. Sobre esta capa funcionan algunas herramientas como por ejemplo HBase, una base de datos columnar que se guarda los datos en distintas carpetas y así saca todo el partido de las características de HDFS.

Sobre la capa HDFS también encontraremos la capa YARN encargada de administrar los recursos de la misma manera que lo hace un planificador de sistema operativo. Desde capas superiores se solicitarán recursos y es esta capa la que debe asignarlos según la demanda y carga de los nodos.

Sobre la capa administradora de recursos está la capa que solicita esos recursos. MapReduce es el modelo con el que nació Hadoop. Hablaremos de este modelo en otro tema, por ahora solo decir que consiste en procesar los datos parciales según estén distribuidos en los nodos y después unir esos resultados parciales en un único resultado final.

Además de MapReduce, hay otros motores de procesamiento como por ejemplo Spark que, aunque mantiene la filosofía de procesar directamente en los nodos, añade mejoras en velocidad al trabajar directamente en memoria.

Hay que tener en cuenta que la mayoría de las herramientas de Hadoop, al menos las que componen su núcleo, están escritas como aplicaciones Java por lo que tienen dependencias con el sistema operativo. Esto quiere decir que Hadoop también acepta herramientas fuera de HDFS y YARN como por ejemplo las herramientas relacionadas con la monitorización.

El modo de funcionamiento de Hadoop cambia respecto a las arquitecturas tradicionales. Hasta ahora conocíamos la arquitectura de John von Neumann donde los datos e instrucciones se guardaban en memorias secundarias hasta el momento de su ejecución que se colocarían en la memoria principal. Las arquitecturas software imitaron esta manera de trabajar y aunque los datos estuvieran distribuidos en varios nodos finalmente acababan en la memoria principal de una sola máquina para su ejecución.

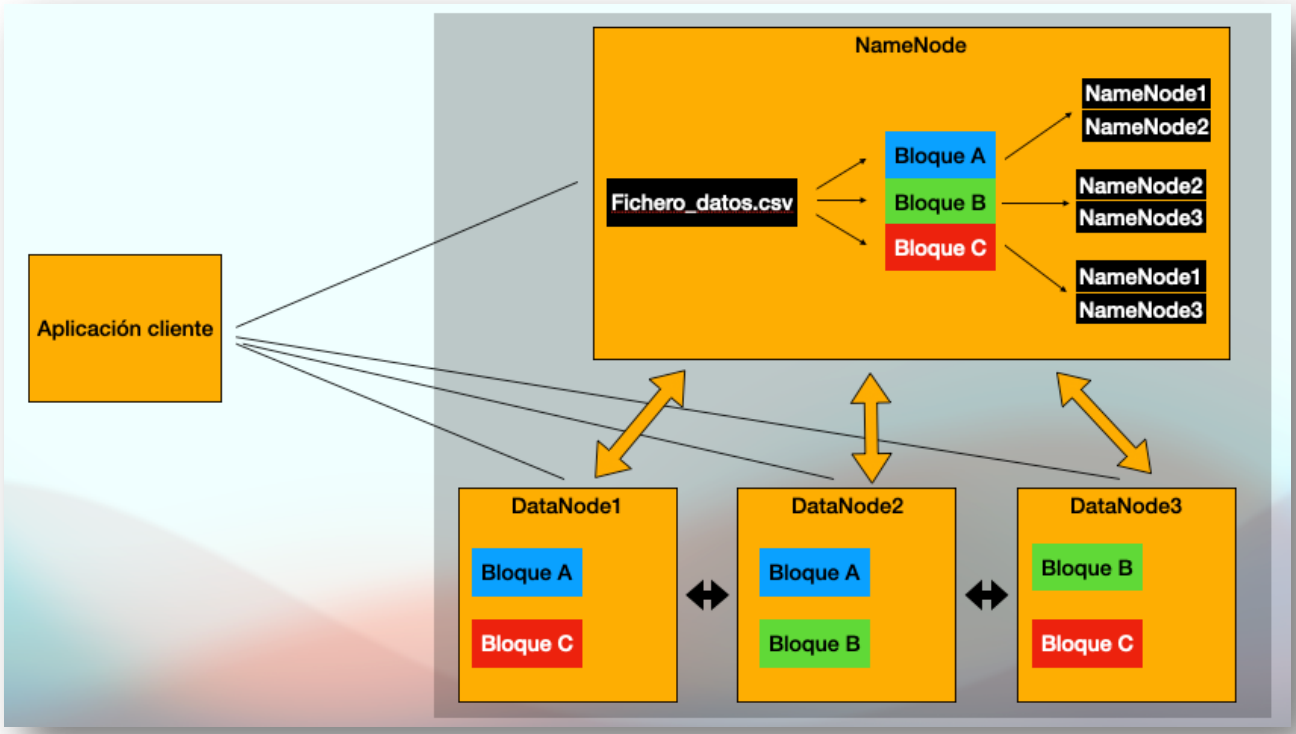
La arquitectura Hadoop supone un cambio importante porque la memoria principal en la que se procesa un dato ya no está en una sola máquina. Ahora puede haber tantas memorias principales para procesar datos como nodos tengamos.

5.- Nodos en un clúster Hadoop

La arquitectura Hadoop desde un punto de vista algo más técnico clasifica a los nodos atendiendo a sus funciones y distinguiendo la parte HDFS de la parte YARN de proceso.

5.1 HDFS

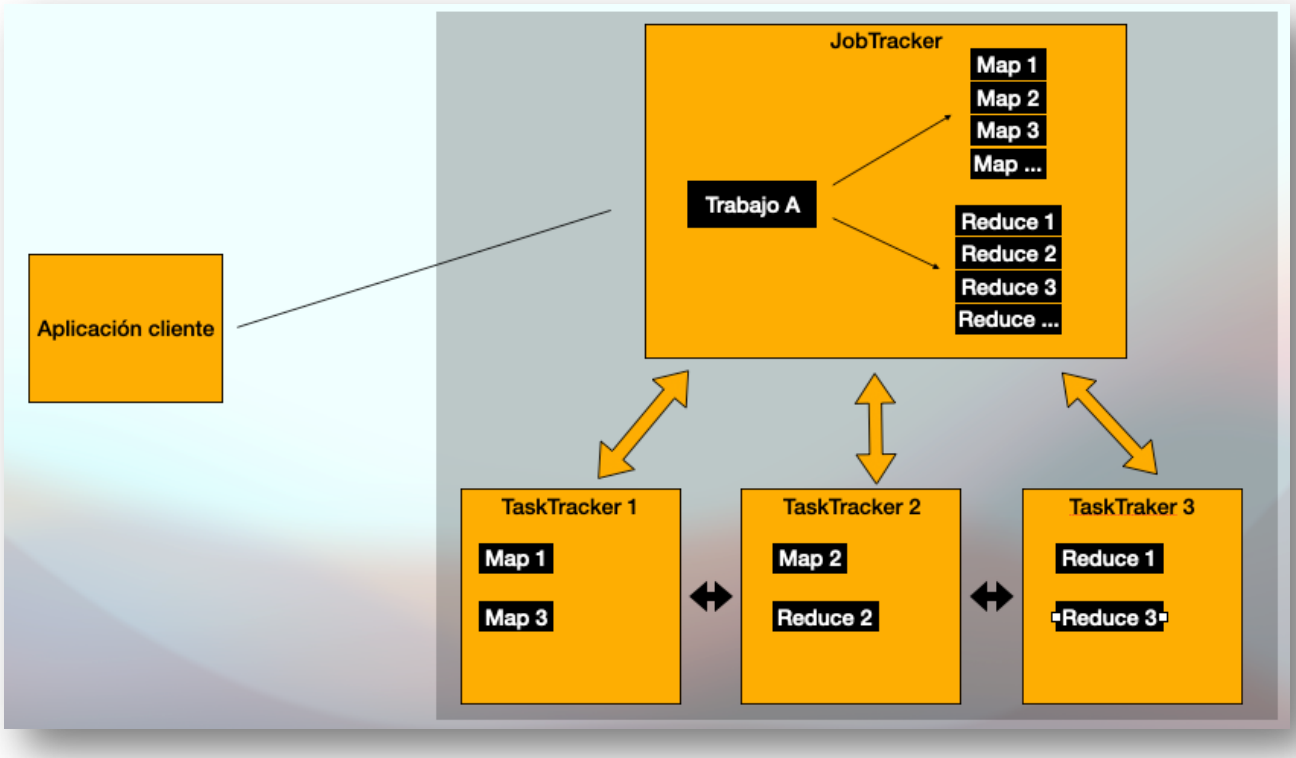
En la parte HDFS encontramos los nodos de almacenamiento (DataNode) y los nodos de gestión del almacenamiento (NameNode).



Los NameNode son los encargados de tener controlada la ubicación de los datos en los distintos nodos. Requiere de un equipo especialmente dotado de memoria RAM para poder gestionar las peticiones lo más rápido posible. Los namenode supone un pequeño cuello de botella ya que los clientes hacen sus peticiones a estos nodos.

Los DataNodes son los nodos que almacenan información en forma de bloques. Estos se comunican entre ellos y con el NameNode para mantener varias réplicas de los bloques y otras tareas como monitorización y servir datos al cliente directamente.

5.2 YARN



En la parte de procesamiento con MapReduce también distinguimos distintos tipos de nodos atendiendo a sus funciones.

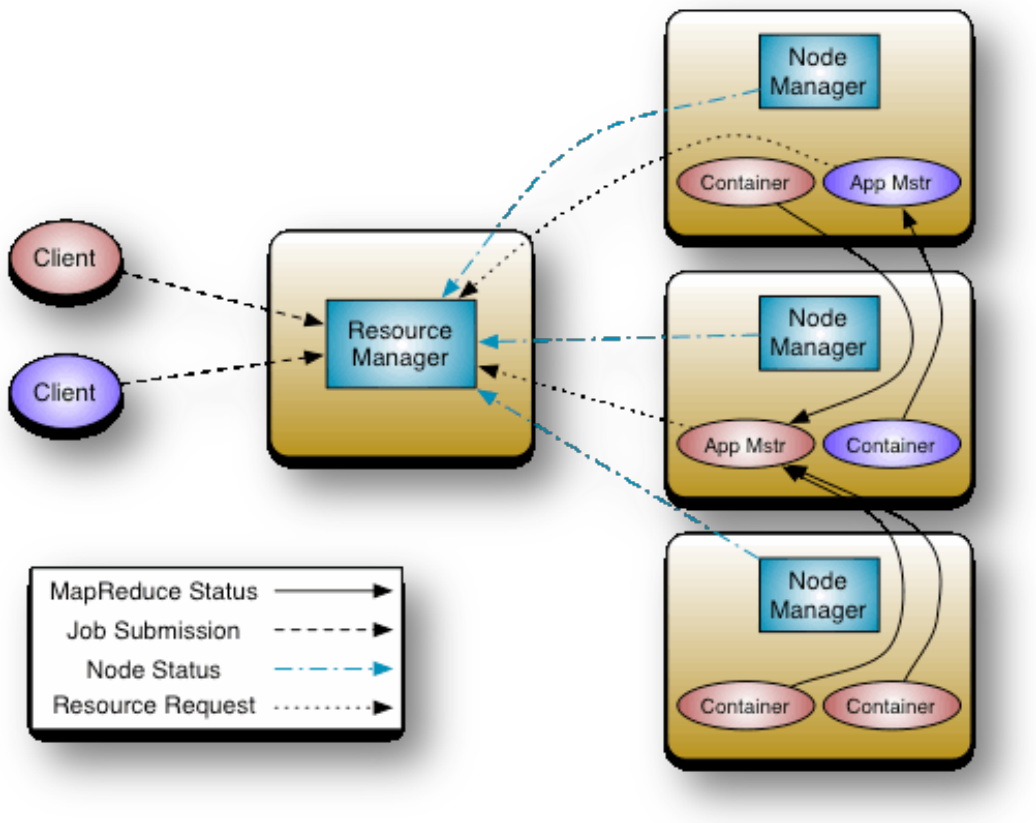
El nodo JobTracker es el que coordina las actividades del resto de nodos esclavos que se dedican al proceso.

Los nodos TaskTracker son los encargados de realizar el proceso. Idealmente realizarán los trabajos Map allí donde se encuentre el bloque. Los procesos Reduce lee los datos parciales a través de la red y devuelve un único resultado como respuesta.

A un nivel más detallado encontraremos 3 componentes:

- Resource Manager. Gestiona los recursos del clúster por lo que suele estar en un namenode de HDFS.
- Node Manager. Gestiona los contenedores que están ejecutando una tarea en ese nodo por lo que suelen estar en los datanodes
- Application Master. Gestiona la ejecución de una tarea concreta y también se aloja en alguno de los datanodes.

El funcionamiento a alto nivel de YARN consistiría en que un cliente lanza una aplicación en YARN. El Resource



Manager tiene dos misiones, por un lado, mediante el scheduler controla el clúster asigna recursos en forma de contenedores con una cantidad de memoria y procesamiento asignado de antemano. Por otro lado, mediante el Application Manager recibe y acepta peticiones de cliente y le asocia su primer contenedor. En caso de que la aplicación fallara sería el encargado de relanzarla.

A continuación, el Application Manager se pone en contacto con los Node Manager. Estos son los encargados de cada nodo y deben informar al Resource Manager de su estado constantemente. Además, son los encargados de crear según las órdenes del Resource Manager y lanzar a ejecución aquellos contenedores que indique el Application Master.

Cada contenedor ejecuta su parte correspondiente. Mientras, el Application Master monitoriza la ejecución e informa tanto al Resource Manager como al Application Manager.

En cuanto la aplicación finaliza el Application Manager informa al Resource Manager y este libera los contenedores.

Dedicaremos un tema a estudiar esta estructura así que por ahora no es necesario preocuparse por tener una idea exacta del funcionamiento.

5.3.- Hardware

En cuanto a la arquitectura desde el punto de vista hardware, Hadoop adopta el conocido como “commodity hardware”. En una interpretación estricta podríamos decir que Hadoop usa hardware no especializado, es decir, equipos servidores son suficientes. Estirando al límite también podremos usar PCs, pero dejando claro que el rendimiento no estará entre las preferencias de uso.

6.- Beneficios, desventajas y dificultades

En muchos escenarios los nodos máster suponen un punto de fallo único y la alta disponibilidad acaba de llegar en la versión 3 de Hadoop.

El nivel de seguridad que tiene Hadoop se basa en el sistema de propiedad y permisos del sistema de ficheros HDFS que, por defecto, viene completamente abierto. Podemos solucionarlo mediante Kerberos que se puede integrar en un dominio haciendo que usuarios y servicio tengan un acceso de tipo “single-sign-on”.

El almacenamiento y las comunicaciones internas del clúster no son encriptadas.

El mayor problema de HDFS es la gestión de archivos pequeños, ausencia de compresión, pobre rendimiento en lecturas aleatorias.

MapReduce es una arquitectura distribuida por lotes y no gestiona bien el tiempo real ni los datos que sean mutables en el tiempo.

Tal vez el problema más grave sea las incompatibilidades entre herramientas del ecosistema Hadoop. Se ha llegado a dar casos en los que una herramienta, Hbase, solo trabajaba con una versión de otra herramienta, HDFS, que aún no había sido verificada para funcionar en producción. Las distintas distribuciones de Hadoop seleccionan y adaptan las herramientas para que puedan trabajar juntas.